

Basit Kamera ve İşaretleyiciler Yardımıyla Dağıtık Formasyon Görev Kontrolcüsü Tasarımı

Distributed Formation Task Controller Design with Simple Camera and Markers

Nurullah Özkan¹, Veysel Gazi²

¹Uçak Mühendisliği Bölümü
Sivas Bilim ve Teknoloji Üniversitesi, Sivas

nozkan@sbtu.edu.tr

² Kontrol ve Otomasyon Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, İstanbul

vgazi@yildiz.edu.tr

Özetçe

Bu bildiride, basit kameralar ve işaretleyiciler kullanılarak verilen görevi yerine getirirken bir İHA sürüsünün formasyon değişikliği yapması ve genişlemesi ve daralması tartışılmaktadır. Çoğu çalışmada, formasyon kontrolünün ana kaygısı, yerel etkileşimleri kullanarak ve merkezi bir karar mekanizmasına ihtiyaç duymadan sürünün şeklini oluşturmak ve korumaktır. Bu çalışmada bunun üstesinden gelmek için İHA'lar arasındaki göreceli konumları tahmin eden yerleşik basit kameralar ve işaretleyiciler kullanılmaktadır. Daha sonra, dağıtılmış kontrolör yasalarımız, formasyon şeklinin ve formasyonun boyutunun değiştirilmesi görevlerini gerçekleştirmek için bu göreceli konumları kullanmaktadır. Bu çalışmanın amacı, İHA'lar arasındaki yerel etkileşimleri kullanarak dağıtılmış bir formasyon kontrolörü oluşturmaktır. Bu yerel etkileşimler, basit kameralardan gelen göreceli konum tahminleri ve ana kontrolör, ajanlar için dağıtılmış kontrolcülerimizdir. Genel olarak bu çalışma, formasyon değiştirme ve yeniden boyutlandırma için bilgisayarla görme tabanlı yaklaşımların potansiyelini göstermek için yapılmış bir ön çalışmadır.

Abstract

This paper discusses the formation change, expansion and contraction of a UAV swarm while performing a given task using simple cameras and markers. In most studies, the main concern of formation control is to establish and maintain the shape of the swarm using local interactions and without the need for a centralised decision mechanism. To overcome this in this work, we use simple on-board cameras and markers that estimate the relative positions between UAVs. Then, our distributed controller laws use these relative positions to perform the tasks of changing the formation shape and the size of the formation.

The goal of this work is to build a distributed formation controller using local interactions between UAVs. These local interactions are the relative position estimates from simple cameras and the master controller, our distributed controllers for agents. Overall, this is a preliminary study to demonstrate the potential of computer vision-based approaches for formation modification and resizing.

1. Giriş

Teknolojinin ilerlemesi ve erişimin kolaylaşmasıyla birlikte İnsansız Hava Araçları (İHA), son yıllarda savunma uygulamalarında sınır koruma ve gözetlemenin yanı sıra tarım, eğlence, ulaşım, yangın algılama, haritalama ve iletişim gibi birçok uygulamada kullanılmaya başlanmıştır. Fakat bazı durumlarda tek bir İHA bir görev için yeterli olmayıp çok sayıda İHA sürü olarak kullanılabilir. Bu gibi bir durum görevin yerine getirilme olasılığının artırılması, tasarımca daha ucuz İHA'larla görevlerin yerine getirilebilmesi gibi birçok avantaja sahiptir.

Sürü davranışlarını oluşturmak için sürü bireylerinin birbirlerine göre sürü içindeki konumlarının bilinmesi gerekmektedir. Bu iki şekilde yerel ve küresel algılama şeklinde yapılabilmektedir. Yerel olanda sürü bireyleri üzerindeki sensörleri kullanarak sürü içindeki diğer bireylerin konumlarını kendine göre tahmin eder. Doğadaki sürülerde buna benzer şekilde komuşu bireyler arasındaki yerel etkileşimleri kullanarak karmaşık sürü davranışları gerçekleştirirler [1]. Küresel algılamaya ise sürü içindeki bireylerin konumlarını tahmin etmek için harici küresel bir konumlayıcının GPS, wifi konumlama sistemleri, UWB veya hareket yakalama sensörleri kullanılması gibi [2] ve bir şekilde (merkez-İHA), (İHA-İHA) tüm sürü bireylerine iletilmesidir. Bu çalışmada kameralar ve basit işaretleyiciler ile yerel algılama yöntemi kullanılarak sürü davranışlarından formasyon kontrolü yapılmıştır.

Çizge teorisi sürü formasyon şekillerini modellemek için kullanılan bir yöntemdir. Özellikle rijit çizge teorisi sürü birey-

leri arasındaki mesafe kısıtlarını göz önünde bulundurarak altında yatan çizgenin katılığından faydalanarak uygulamaya çalışır [3]. Ayrıca rijitlik sürü bireyleri arasında oluşabilecek çarpışmaları önlemesini de sağlar. Formasyon kontrolünde sürü bireyleri arasındaki mesafenin kontrol edilmesi ayrıca küresel bir konumlama sisteminin gerekliliğini ortadan kaldırarak [4] sürü bireylerinin yerel algılama sistemlerini kullanarak dağıtık bir sürü formasyon kontrolü yapılmasına olanak sağlar. Çizge teorisinin sürü formasyonunda kullanılması ile ilgili verilen çalışmaya bakılabilir [5].

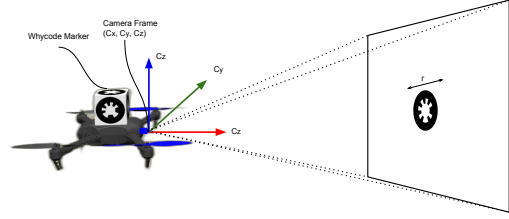
İHA sürüleri formasyon halinde bir görevi yerine getiren ortam şartlarına bağlı olarak zamanla ilk baştaki verilen formasyon şeklini değiştirmeleri veya görevlerine bağlı olarak formasyon şeklini küçültüp büyütme gerekebilmektedir. Bu çalışmada bizim amacımız sürü formasyonunun şeklinin görev gerekliliklerine bağlı olarak değiştirilmesi ve yeniden boyutlandırılmasıdır. Bu çalışma metodoloji, simülasyon sonuçları ve sonuçlar kısmından oluşmaktadır.

2. Metodoloji

Bu kısımda genel metodolojiden bahsedilmektedir. İstenilen formasyon şeklinin oluşturulması ajanlar arasındaki relatif mesafelerin kontrolü ve buda relatif pozisyonlarının tahminine bağlı olarak yapılmaktadır. Bu tahminler sonrasında dağıtık kontrolcüler tarafından kullanılarak istenilen formasyon şeklini oluşturulmaktadır. Çoğu çalışmada basit dinamikli ajan dinamikleri üst seviye kontrolcü olarak sürü formasyon veya koordinasyon kontrolcüsü olarak kullanılırken; alt seviye kontrolcü ajanın yani İHA'nın kontrolünden sorumlu olmaktadır [6]. Fakat bu sürü kontrolünün İHA'lar üzerine uygulanması için İHA'ların ulaşılması istenilen pozisyon bilgilerine dönüştürmeleri gerekmektedir [7]. Bu çalışmada tek dinamikli ajan modellerini kullanarak üst seviye sürü kontrolcülerini oluşturulmuş ve çıktıları sonrasında alt seviye kontrolcüye uygun girdi olarak verilmiştir.

2.1. Relatif Konum Tahmini

Bu çalışmada, sürü bireyleri arasında herhangi bir iletişim ihtiyacı olmadan sadece kendi kamerası ile gözlemlediği işaretleyicilerin konumlarından yola çıkarak kamerası görüntüsü üzerindeki İHA'ların relatif konumları tahmin edilmektedir. Bu tahmin özellikle ulaşılması kolay olan basit kameralar ve beyaz-siyah işaretleyiciler kullanılarak yapılmaktadır. Konum tahmini amaçlarla kullanılan bir çok işaretleyici, örnek olarak en çok bilinenler Aruco, AprilTag, ArTag, whycode, and whycone [8] tasarımı bulunmaktadır. Bu çalışmada whycode işaretleyicilerinin karmaşıklığının daha az olması, böylece gerçek ortam testlerinde daha az işlem gücüne ihtiyaç duyması ve işaretleyicilerin boyutları küçük olsa bile algılama mesafesi fazla olması gibi avantajlarından dolayı tercih edilmiştir [9]. Relatif konum tahmini için Whycode işaretleyicilerin anlaşılması gerekmektedir. Bunun için geliştirilen açık kaynaklı bir kütüphane kullanılmıştır [9]. Gazebo'dan alınmış Şekil 1'de, kameralı İHA ve işaretleyiciden oluşan sistemin örnek bir şeması gösterilmektedir.



Şekil 1: İris İHA üzerine monte edilmiş kamera ve markerlarda aluşan sistemin örnek bir şeması.

2.2. Sürü Bireylerinin Hareket Modeli

Formasyon kontrolü amacıyla, her bir İHA için dağıtık kontrolcüler kinematik ajan modeli üzerine tasarlanmıştır. Bu kontrolcüler üst seviye sürü kontrolcüleridir ve alt seviye kontrolcü için gerekli referans girdilerin üretilmesinde kullanılır. Bir nevi bizim çalışmamız açısından yol planlayıcı görevi görür. Sürümüz $N \geq 3$ ajandan oluşmakta ve $S = A_1, \dots, A_N$ ile temsil edilmektedir. Her bir ajan A_1, \dots, A_N üç boyutlu uzayda (R^3), (1) denkleminde verildiği gibi her yöne serbestçe hareket edebilirler.

$$\dot{p}_i(t) = u_i(t), \quad i = 1, 2, \dots, N. \quad (1)$$

Bu denklemden, $p_i(t) = (x_i(t), y_i(t), z_i(t))$ ve $u_i(t)$, i 'inci ajanın konumunu ve kontrol girdisini/hızını temsil etmektedir.

2.2.1. Formasyonun Modellenmesi

Çizge teorisi çok ajanlı formasyon, algılama, iletişim ve kontrol yaklaşımlarının modellenmesi için yaygın olarak kullanılan bir araçtır. Bu çalışmada formasyonun sanal yapısını ifade etmek için çizge teorisindeki minimal katı yönlendirilmiş çizgelerden faydalanılmaktadır. Katı bir formasyon elde edebilmek için mesafe kısıtlarının sağlanması gerekmektedir. Bu şekilde kalıcı bir formasyon modeli oluşturulmuş olmaktadır. Sürü S için formasyon modeli $C = (D, K)$ şeklinde yönlendirilmiş bir çizge olarak ifade edilirken, D , S içindeki ajanları düğüm olarak ve K yönlendirilmiş kenar kümelerini temsil eder. Yönlendirilmiş kümeler araçlar arasındaki bilgi akışını ve yönünü temsil ederken düğümler ajanları yani İHA'ları temsil etmektedir. Örnek olarak bilgi akışı yerleşik basit kamerası olan A_j İHA'sının yerleşik bir işaretleyicisi olan A_i 'yi tespit etmesi ve kendi referans merkezine göre m_{ij} relatif mesafesidir. A_j 'nin amacı formasyon şekline bağlı olan m_{ij} mesafesini diğer formasyon hedeflerini göz önünde bulundurarak ayarlamaktır. Burada m_{ij} mesafesi A_j 'nin A_i 'yi takip etmesi için gerekli referans girdisidir. Bu çalışma için bu girdi formasyon değiştirme görevine bağlı olarak değişebilmekte ve formasyonun büyütülüp küçültülmesi için zamana bağlı olarak tanımlanabilmektedir. Genel olarak ifade edilmek istenirse formasyon şekli için istenilen mesafeler $M_f^k(t) = \{m_{f^{kj}}(t) | (i, j) \in K\}$ şeklinde ifade edilebilir. Burada t zamanla bir değişim olarsa dahil edilir. f^k formasyon şeklini ifade etmektedir ve k da mevcut şekilleri temsil etmektedir. Örnek olarak herhangi bir t anında formasyon değiştirme komutu geldiğinde $M_f^{k=1}(t)$ ile $M_f^{k=2}(t)$ yer değiştirip oluşturulmak istenen yeni formasyon şekline uygun yeni bir kenar kümesi K tanımlanmalıdır. Tüm bilgileri göz önüne alarak is-

tenilan formasyon şekli $F = (S, C, M_f(t))$ olarak temsil edilmektedir. Bu çalışmada, formasyon görevlerini yerine getirebilmek için lider-takipçi tabanlı bir yöntem kullanılmaktadır [10]. Sürü içinde bir lider, bir eş lider ve takipçiler olmak üzere üç farklı sınıf yer almaktadır. Ayrıca formasyonlar 2B düzlemsel formasyonlardır.

2.3. Dağıtık Kontrol Yöntemi

Bu bölümde, formasyon modellemesinde ulaşılmak istenen hedeflere uygun kontrolcüler verilmektedir. Uyumlu hareket elde etmek için, bireysel kontrolörler ilk olarak minimum kalıcı formasyonu sürdürmeli, formasyon değiştirme komutu geldiğinde bunu sağlamalı, istenirse formasyon şeklinin boyutunu değiştirebilir olmalı ve aynı zamanda bir gezingeyi veya ara noktaları takip etmelidir. Burada herhangi bir global veriye ihtiyaç duymayan bir sürü sistemi yaratılmak istendiği için bireysel kontrolörler ajan kordinat sistemine dayandırılmalıdır. Bunu anlamlandırmak için, ajan A_i 'nin ajan A_j 'nin koordinat merkezine göre konumunu/relatif pozisyonu $p_{ij}^i(t) = p_i(t) - p_j(t)$ 'dir. Bu çalışmada relative pozisyon dronlar üzerinde ki kemeralar ve marker ile tahmin edilmiştir. Herhangi bir t zamanında kendi koordinat eksenindeki istenen son konumunu $p_{if}^i = p_{if} - p_i(t)$ ve kendi konumu $[0, 0, 0]^T$ 'dir. Aşağıda taratılan kontrol çıktıları, ajanlar ne kadar R^3 'de hareket etsede 2B formasyon oluşturduğundan dolayı $u_i(t) = [u_{ix}(t), u_{iy}(t), 0]^T$ şeklindedir. İHA'lar alt seviye kontrolcü ile irtifalarını kontrol etmektedirler.

2.3.1. Liderin Kontrolcüsü

Lider-takipçi formasyon kontrol yaklaşımında, lider ajan A_1 'in tek görevi ya bir yörüngeyi takip etmek ya da istenen konuma veya konumlara gitmektir. Liderin hız girdisi, son konum (veya yörünge) $p_{1f}^{(1)}$ yönünde olacak şekildedir. Lider A_1 için kontrol yasası (2) olarak hesaplanır.

$$u_1(t) = \sigma_1 \bar{v} \bar{\beta}_1(t) p_{1f}^{(1)}(t) / \|p_{1f}^{(1)}(t)\| \quad (2)$$

$$\bar{\beta}_1(t) = \begin{cases} 0, & \|p_{1f}^{(1)}(t)\| < \epsilon_f \\ \frac{\|p_{1f}^{(1)}(t)\| - \epsilon_f}{\epsilon_f}, & \epsilon_f \leq \|p_{1f}^{(1)}(t)\| < 2\epsilon_f \\ 1, & \|p_{1f}^{(1)}(t)\| \geq 2\epsilon_f \end{cases} \quad (3)$$

Eşitlik (2)'deki $0 < \sigma_1 < 1$, A_1 'in çok hızlı hareket etmesini önlemek ve $A_k | \overrightarrow{(k, 1)}$ takipçileri için izlemeyi mümkün kılmak için kullanılan önceden ayarlanmış bir sabittir. \bar{v} tüm ajanlar için önceden tanımlanmış pozitif maksimum bir hızı temsil etmektedir. Eşitlik (3)'deki $\bar{\beta}_1(t)$, A_1 'in son konumundaki küçük hatalardan kaynaklanabilecek çaptırdamaları önlemek için kullanılmaktadır. ϵ_f yavaşlama ve durma aralıklarını temsil eden önceden belirlenmiş küçük pozitif bir sabittir.

2.3.2. İkinci Liderin Kontrolcüsü

A_2 eş lider olarak kabul edilir. A_2 'nin görevleri lideri takip etmek ve arasındaki mesafe kısıtını sağlamak ve ulaşması istenen konuma doğru hareket etmektir. Bunu mümkün kılmak için, lider ve p_{2f} eğer ulaşılması istenen noktaya görüntü işleme ile ulaşıp algılanacaksa her zaman yardımcı liderlerin kamerasının görüş alanında olmalıdır. Bazı durumlarda, A_2 lideri ve ulaşması istenen konumu aynı anda tespit edemezse, hem lider

A_1 üzerindeki işaretleyiciyi hem de son konumu p_{2f} görene kadar lider A_1 etrafında dönmesi gerekir. Bu çalışma için işaretleyicinin kimliği 1'dir. Eş lider A_2 için kontrol yasası eşitlik (4) olarak verilmiştir.

$$u_2(t) = \beta_2(t) u_{21}(t) + \sqrt{1 - \beta_2^2(t)} u_{22}(t) \quad (4)$$

$$\beta_2(t) = \begin{cases} 0, & |\bar{\delta}_{12}(t)| < \epsilon_{ik} \\ \frac{|\bar{\delta}_{12}(t)| - \epsilon_{ik}}{\epsilon_{ik}}, & \epsilon_{ik} \leq |\bar{\delta}_{12}(t)| < 2\epsilon_{ik} \\ 1, & |\bar{\delta}_{12}(t)| \geq 2\epsilon_{ik} \end{cases} \quad (5)$$

Eşitlik (4)'de görülebileceği gibi, eş liderin girdisini bulmak için iki girdinin toplanması gerekmektedir. Bu girdilerden $u_{21}(t)$, A_1 'e formasyonun şeklinin kısıtına bağlı olarak istenen mesafede sabit tutmaya çalışırken aynı zamanda eş liderin lideri ve liderin yönünü takip etmesini sağlayan girdidir. u_{22} lideri ve istenen konumu aynı anda görmesini sağlayacak şekilde hareket etmesini sağlayan girdidir ve formasyon şeklini yönlendirmek için kullanılmaktadır. Eşitlik (5)'daki ϵ_{ik} küçük pozitif bir sabittir. $\beta_2(t)$ girdilerin baskınlığını ayarlamak için kullanılmaktadır.

$$u_{21}(t) = \sigma_2 \bar{v} \text{sgn}(\bar{\delta}_{12}(t)) p_1^{(2)}(t) / \|p_1^{(2)}(t)\| \quad (6)$$

$$u_{22}(t) = \begin{cases} \sigma_2 \bar{v} \bar{\delta}_{12}^\perp(t), & \text{if } ID \neq 1 \\ \sigma_2 \bar{v} \bar{\beta}_2 \text{sgn}(p_{2f}^{(2)T} \bar{\delta}_{12}^\perp(t)) \bar{\delta}_{12}^\perp(t), & \text{if } ID = 1 \end{cases} \quad (7)$$

$$\bar{\delta}_{12}(t) = \|p_1^{(2)}(t)\|^2 - m_{fk12}^2(t) \quad (8)$$

$$\bar{\delta}_{12}^\perp(t) = (-p_{1y}^2(t), p_{1x}^2(t)) / \|p_1^{(2)}(t)\| \quad (9)$$

$$\bar{\beta}_2(t) = \begin{cases} 0, & |p_{2f}^{(2)}(t)| < \epsilon_f \\ \frac{|p_{2f}^{(2)}(t)| - \epsilon_f}{\epsilon_f}, & \epsilon_f \leq |p_{2f}^{(2)}(t)| < 2\epsilon_f \\ 1, & |p_{2f}^{(2)}(t)| \geq 2\epsilon_f \end{cases} \quad (10)$$

Eşitlik (6) ve (7)'deki $0 < \sigma_2 < 1$ takipçiler için takip etmeyi mümkün kılmak için kullanılan önceden ayarlanmış bir sabittir. Eşitlik (8)'da ki $\bar{\delta}_{12}$ lider ve eş lider arasındaki mesafeyi kontrol etmek için kullanılır. $m_{fk12}(t)$ lider ile eş lider arasındaki formasyon görevine bağlı mesafeyi temsil etmektedir. Eşitlik (9)'daki $\bar{\delta}_{12}^\perp(t)$ lider ve takipçi arasındaki mesafeye dik olan vektörü temsil etmektedir. Eşitlik (10)'daki ϵ_f küçük pozitif bir sabittir ve yavaşlama ve durma aralıklarını temsil etmektedir. $\bar{\beta}_2(t)$ ulaşması istenen konumuna doğru hareket ederken küçük hatalardan kaynaklanabilecek çaptırdamaları önlemeye çalışmaktadır.

2.3.3. Takipçilerin Kontrolcüsü

A_i , $i = 3, 4, \dots, N$ takipçi ajandır. A_i 'nin görevi formasyon modellemesinde belirlenmiş iki ajanı formasyon isterlerine bağlı olacak şekilde takip etmektir. Örnek olarak lider $A_{j=1}$ ve eş lider $A_{l=2}$ 'yi şeklinde tanımlanırsa A_i takipçisi lider ve eş lideri takip etmek için; lider A_i 'i, $\|p_{id}^{(i)}(t) - p_k^{(i)}(t)\| = m_{fkij}(t)$ ve eş lider A_2 'yi $\|p_{id}^{(i)}(t) - p_j^{(i)}(t)\| = m_{fkil}$ koşullarını sağlayacak şekilde $p_{id}^{(i)}(t)$ noktasına doğru hareket etmelidir. $p_{id}^{(i)}(t)$, $C(p_k^{(i)}(t), m_{fkij}(t))$ ve $C(p_j^{(i)}(t), m_{fkil}(t))$ dairelerinin kesişimi en yakın nokta olarak hesaplanmaktadır. Daireler $C(p, d)$

biçimindedir; burada p merkezin konumu ve d dairenin yarıçapıdır. Daha fazla takipçi varsa, bunlar A_3 'e benzer şekilde tasarlanabilmektedir. Takipçiler için kontrol yasası eşitlik (11) olarak verilmektedir.

$$u_i(t) = \sigma_i \bar{v} \beta_i(t) p_{id}^{(i)}(t) / \|p_{id}^{(i)}(t)\| \quad (11)$$

$$\bar{\beta}_i(t) = \begin{cases} 0, & \|p_{id}^{(i)}(t)\| < \epsilon_{tk} \\ \frac{\|p_{id}^{(i)}(t)\| - \epsilon_{tk}}{\epsilon_{tk}}, & \epsilon_{tk} \leq \|p_{id}^{(i)}(t)\| < 2\epsilon_{tk} \\ 1, & \|p_{id}^{(i)}(t)\| \geq 2\epsilon_{tk} \end{cases} \quad (12)$$

Eşitlik (12)'deki $0 < \sigma_i \leq 1$ takipçilerin ulaşması istenen konumdaki p_{id} oluşabilecek değişikliklerin takibi mümkün kılacak kadar küçük olmasını sağlamak için kullanılan bir ön ayar sabitidir. ϵ_{tk} küçük pozitif bir sabittir ve ulaşılması istenen konuma ulaşırkenki yavaşlama ve durma aralıklarını temsil etmektedir. $\bar{\beta}_i(t)$ sabit tutulması istenen ajanlar arası mesafelerdeki olası küçük hatalar nedeniyle oluşabilecek çatırdamaları önler. Takipçilerin kontrol yasasında $p_{id}^{(i)}(t)$ değerlerini hesaplamak için kullanılan $p_j^{(i)}(t)$ ve $p_k^{(i)}(t)$ kameralar ve whycode işaretleyicileri kullanılarak hesaplanmaktadır be bunlar relatif pozisyonlardır.

2.4. Dönüşüm Kuralı

Ajanlar için oluşturulmuş kontrolçüler ajanların basit dinamik modelde hareket ettiği varsayılarak oluşturulmaktadır fakat İHA'ların yani alt seviye kontrolcünün anlayıp uygulayabileceği, ulaşmak istenen pozisyon girdilerine dönüştürmeleri gerekmektedir. Bu da [7] çalışmasında da olduğu gibi eşitlik (13) benzer şekilde yapılmaktadır.

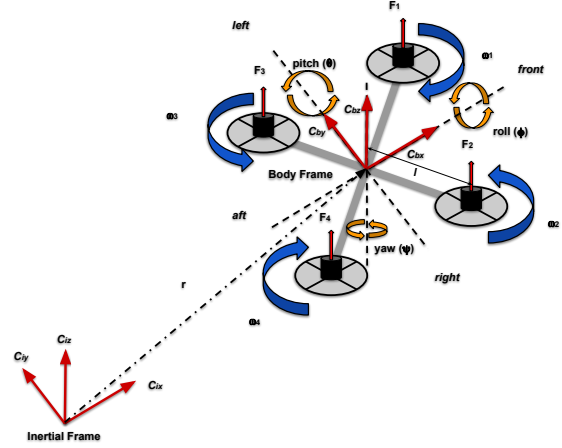
$$p_i^d(c+1) = p_i^d(c) + T u_i(c) \quad (13)$$

Burada T örnekleme periyodu ve c mevcut zamandır. p_i^d İHA'nın ulaşılması istenen konumu temsil etmektedir.

2.5. İHA'nın Dinamiği ve Kontrolü

2.5.1. İHA Dinamiği

Bu çalışma için İHA'lar bir drondur. Dron eksik tahrikli, dört pervaneli bir hava aracıdır. Dronun matematiksel modelinin nasıl oluşturulduğu burada ayrıntılı bir şekilde verilmemektedir. Ayrıntılarına [7, 11] kaynaklarından ulaşılabilir. Burada modellenen dron Şekil 2 görselleştirildiği gibidir. $r^T = [x, y, z]$ dronun eylemsiz koordinat sistemindeki konumunu temsil etmektedir. Dronun yönelimi, Euler açıları $\Omega^T = [\phi, \theta, \psi]$ sırasıyla pitch, roll ve yaw açıları olarak verilir. Dronun eylemsiz koordinat sisteminde modellenmesi plotun veya yerden bakan birisinin anlayabilmesi açısından gereklidir. Bunun içinde dronun gövde koordinatlarının eylemsiz koordinatlara dönüştürülmesi gerekmektedir. Rotorlar $i = 1, 2, 3, 4$ tarafından üretilen F_i itme kuvveti $F_i = b w_i^2$ şeklindedir. b itme faktörü, m quadcopterin kütlesi ve w_i rotor i 'nin açıl hızıdır (rad/s). Ayrıca, rotorlar gövde eksenini etrafında dronun yönelimini sağlayan $\tau^T = [\tau_\phi, \tau_\theta, \tau_\psi]$ şeklinde momentler oluşturur. $\tau_\phi = l_x b (-w_1^2 + w_2^2 - w_3^2 + w_4^2)$, $\tau_\theta = l_y b (w_1^2 + w_2^2 - w_3^2 - w_4^2)$ ve $\tau_\psi = d (w_1^2 - w_2^2 - w_3^2 + w_4^2)$ b itki d sürüklenme faktörleridir. X yapılı bir dron için l kütle merkezi ile rotor arasındaki mesafeyi, l_x ve l_y , C_{bx} ve C_{by} eksenlerine olan mesafeleri temsil



Şekil 2: Dronun konfigürasyonu.

etmekte ve $l_x = l_y = l/\sqrt{2}$ şeklinde hesaplanmaktadır. Dronun C_{ix} , C_{iy} ve C_{iz} yönlerindeki hareketi denklem (14) ile rotasyonel hareketi ise denklem (15) ile matematiksel olarak modellenmektedir.

$$\ddot{r}m = -mge + R \sum_{i=1}^4 F_i e \quad (14)$$

$$J \ddot{\Omega} = -\dot{\Omega} \times J \dot{\Omega} - \sum_{i=1}^4 J_r (\dot{\Omega} \times e) w_i + \tau \quad (15)$$

R rotasyon matrisidir ve koordinat dönüşümü için kullanılmaktadır. Eylemsizlik matrisi J bir köşe matristir ve J_x , J_y ve J_z 'den oluşmaktadır. J_r rotor eylemsizliğidir. Motorların dönüşü asıl dronun hareket etmesini sağlayan kısımlardır, ancak elde edilen modele dayanarak yeni yapay girdiler $U^T = [U_1, U_2, U_3, U_4]$ dönüştürülebilmektedir. Daha sonrasında dron kontrolünde kullanabilmek için motor girdilerine tekrardan dönüştürülmesi gerekmektedir.

2.5.2. İHA Kontrolü

Dron kontrolü genellikle iç döngünün araç kontrol döngüsü ve dış döngünün görev kontrol döngüsü olduğu iç içe kontrol döngülerine sahiptir. Kontrolçüler tasarlanırken PD kontrolcülerinden yararlanılmıştır ve K_p ve K_d 'ler sırasıyla oransal ve türevsel kontrol katsayılarıdır. Araç kontrol döngüsü, dronun havadaki rotasyonunu kontrol etmek için eşitlik (16) gibi tasarlanmaktadır [7].

$$U_2 = (u_\phi + u_\phi^{pd}) \frac{J_x}{l}, U_3 = (u_\theta + u_\theta^{pd}) \frac{J_y}{l}, U_4 = (u_\psi + u_\psi^{pd}) J_z, \quad (16)$$

$u_\phi := \dot{\theta} \dot{\psi} J_1 - \frac{J_B}{J_x} \dot{\theta} w_r$, $u_\theta := \dot{\phi} \dot{\psi} J_2 + \frac{J_B}{J_y} \dot{\phi} w_r$ ve $u_\psi := \dot{\phi} \dot{\psi} J_3$ dronun dinamik denklemlerinden elde edilen ifadelerdir. J_1 , J_2 ve J_3 eylemsizlik momenti kısaltmalarıdır ve $J_1 = \frac{J_y - J_z}{J_x}$, $J_2 = \frac{J_z - J_x}{J_y}$ ve $J_3 = \frac{J_x - J_y}{J_z}$ şeklinde hesaplanabilir. J_r rotorun eylemsizliği, w_r bağlı hızdır ve $w_r = -w_1 + w_2 + w_3 - w_4$ olarak hesaplanır. J_r ve w_r rotorların neden olduğu jiroskopik etki ile ilgilidir. $u_\phi^{pd} = K_{p,\phi}(\phi_d - \phi) + K_{d,\phi}(\dot{\phi}_d - \dot{\phi})$, $u_\theta^{pd} = K_{p,\theta}(\theta_d - \theta) + K_{d,\theta}(\dot{\theta}_d - \dot{\theta})$ ve $u_\psi^{pd} = K_{p,\psi}(\psi_d - \psi) + K_{d,\psi}(\dot{\psi}_d - \dot{\psi})$ PD kontrolcüsü kullanılarak hesaplanan

yardımcı girdilerdir. ϕ_d, θ_d ve ψ_d istenen yuvarlanma, yunuslama ve sapma açılarıdır.

Görev kontrolörü, irtifa kontrolörü ve konum kontrolörü olarak ikiye ayırabilmektedir. İrtifa kontrolörü, dronun eylemsiz ekseninde yerden yüksekliğini hesaplamak için kullanılmaktadır. İrtifa kontrolörü, eşitlik (17) gibi tasarlanmaktadır. İrtifa kontrolcüsü tasarlanırken $\cos(\theta)\cos(\phi) \neq 0$ varsayımı yapılmış ve $u_z = g$ yerçekimi ivmesini kompanse edilecek şekilde verilmiştir. Yardımcı input $u_z^{pd} = K_{p,z}(z_d - z) + K_{d,z}(\dot{z}_d - \dot{z})$, PD kontrolcüsü kullanılarak hesaplanmaktadır. z_d istenen irtifadır.

$$u_1 = u_z + u_z^{pd} \frac{m}{\cos(\theta)\cos(\phi)} \quad (17)$$

Konum kontrolörü, dronun eylemsiz ekseninde yapacağı yanal hareketlerini kontrol etmek için PD kontrolcü olarak tasarlanmıştır. $u_x^{pd} = K_{p,x}(x_d - x) + K_{d,x}(\dot{x}_d - \dot{x})$ ve $u_y^{pd} = K_{p,y}(y_d - y) + K_{d,y}(\dot{y}_d - \dot{y})$ yanal hareketleri kontrol etmektedir. x_d ve y_d üst seviye sürü kontrolcüsünden gelecek ulaşılması istenen konumlarıdır. Eğer bir dron havada asılı iken $\phi_d = \frac{1}{g}(u_x^{pd} \sin(\psi) - u_y^{pd} \cos(\psi))$ ve $\theta_d = \frac{1}{g}(u_x^{pd} \cos(\psi) + u_y^{pd} \sin(\psi))$ bu şekilde hesaplanabilmektedir. İstenilen sapma açısı da üst seviye kontrol çıktılarında $\psi_d = \tan(u_{iy}/u_{ix})^{-1}$ şeklinde hesaplanmaktadır.

Dron dinamikleri alt bölümünde bahsedildiği gibi, kontrol girdileri hesaplanırken $U^T = [U_1, U_2, U_3, U_4]$ yapay kontrol girdilerine dönüştürülmekteydi ancak yapay girdiler quadcopteri kontrol etmek için uygun değildir; bu nedenle, eşitlik (18) kullanarak bunları rotorların ulaşması istenen dönme hızlarına dönüştürmemiz gerekmektedir.

$$\begin{bmatrix} w_1^{des} \\ w_2^{des} \\ w_3^{des} \\ w_4^{des} \end{bmatrix} = \frac{1}{b} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -l_x & l_x & -l_x & l_x \\ l_y & l_y & -l_y & -l_y \\ k & -k & -k & k \end{bmatrix}^{-1} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (18)$$

$k = \frac{b}{d}$ 'dir. Burada istenen motor hızlarının 0 ile w_{max} arasında olmasının sağlaması gerekmektedir. w_{max} motorun maksimum hızıdır. Elde edilen istenen motor dönüş hızları, motor dinamikleri için girdi olarak kullanılmaktadır. Motor dinamikleri eşitlik (19) gibi yazılabilir. Burada w_i motorun dönüş hızı, w_i^{des} istenen motor dönüş hızı ve τ_m motor katsayısıdır.

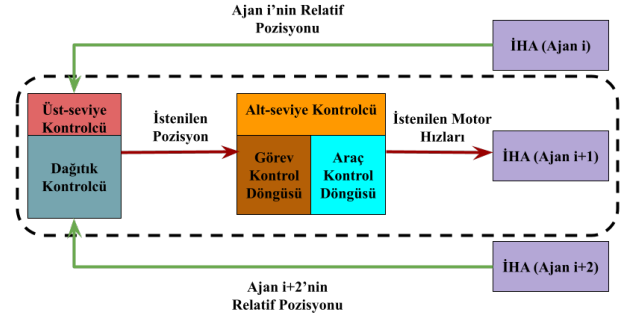
$$\dot{w}_i = \tau_m (w_i^{des} - w_i) \quad i = 1, 2, 3, 4 \quad (19)$$

2.6. Üst Seviye ve Alt Seviye Kontrolcü Bağlantısı

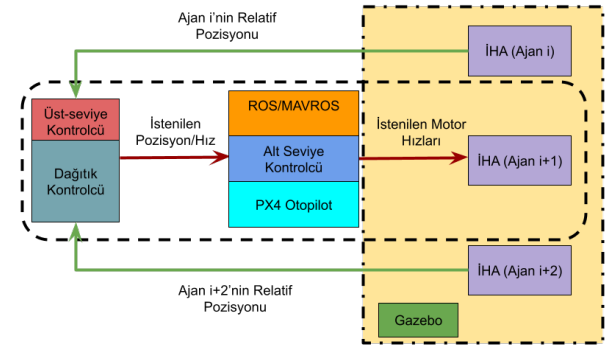
Şekil 3 te görüleceği gibi, üst seviye kontrolcü yani dağıtık sürü kontrolcü çıktıları, alt seviye dron kontrolcüsünün anlayabileceği şekilde ulaşılması istenen pozisyon olarak dönüştürülüp alt seviye kontrolcüye girdi olarak verilmektedir. Buradan da anlaşılacağı gibi aslında üst seviye kontrolcü bir nevi yol planlayıcısı olarak çalışmaktadır.

2.7. Daha Gerçekçi Simülasyon Ortamı

Bu kısımda Gazebo'da PX4 otopilotu kullanan iris dronlar üzerine ROS/MAVROS komutları kullanılarak OFFBOARD modda python üzerinden kontrol yapılmıştır. Burada yapılan testler gerçek ortamda da benzer mantıkla çalışabilecek dronlar üzerinde oluşturulabilecek senaryoların daha gerçekçi sonuçlarını vermektedir. Burada ki sürü sistemi kontrolü Şekil 4'de verildiği gibidir.



Şekil 3: Formasyon görevlerini yerine getiren sürü sisteminin kontrol şeması.



Şekil 4: Formasyon görevlerini yerine getiren sürü sisteminin daha gerçekçi simülasyon ortamındaki kontrol şeması.

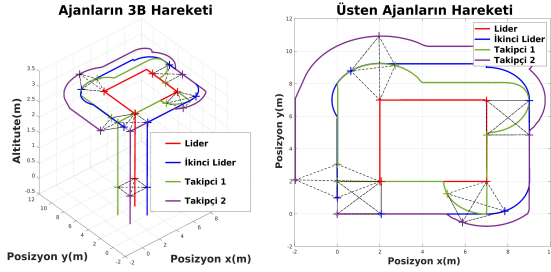
3. Simülasyon Sonuçları

Burada simülasyonlar, ilk olarak MATLAB'ta Şekil 3'te verilen kontrol şemasına uygun olarak ikinci olarak Gazebo, PX4, ROS/MAVROS ve Python'ı kullanarak Şekil 4'te verilen kontrol şemasına uygun olarak gerçekleştirilmiştir. Simülasyonlarda \bar{v} , σ_1 , σ_2 , $\sigma_{3,4,\dots,N}$, ϵ_f , ϵ_{ik} ve ϵ_{tk} değerleri MATLAB'ta 4, 0.68, 0.68, 1 0.02, 0.05 and 0.07 ve Gazebo'da 0.4, 0.5, 0.9, 0.9, 0.01, 0.1 and 0.01 şeklinde seçilmiştir.

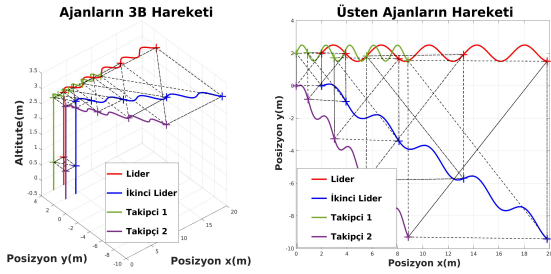
3.1. MATLAB Simülasyonları

MATLAB simülasyonlarında dron kontrolörü için kullandığımız oransal $K_{p,\phi}$, $K_{p,\theta}$ ve $K_{p,z}$ ve türevsel $K_{d,\phi}$, $K_{d,\theta}$ ve $K_{d,z}$ kazançlar sırasıyla 10 ve 6'dır. $K_{d,\psi}$ ve $K_{d,\psi}$ sırasıyla 12 ve 15 tir. Konum kontrolörü için oransal $K_{p,x}$ ve $K_{p,y}$ ve türev $K_{d,x}$ ve $K_{d,y}$ kazançları sırasıyla 2 ve 3'tür. MATLAB'ta craziflie modeli için fiziksel özellikler kullanılmıştır. Bunlar $m = 0.033kg$, $L = 39.73e - 3m$, $J_x = 1.395e - 5kgm^2$, $J_y = 1.436e - 5kgm^2$, $J_z = 2.173e - 5kgm^2$, $b = 2.8799e - 08$ ve $d = 7.2385e - 10$, $w_{max} = 3000rad/s$, $g = 9.81 \frac{m}{s^2}$ ve $\tau_m = 20s^{-1}$ değerleridir. Simülasyonlarda dronlar öncelikle 3m yüksekliğe başlangıç formasyonunu koruyarak çıkar sonrasında bir formasyon değişikliği komuta gelinceye kadar formasyonunu koruyarak görevine devam eder. Komut geldikten sonra yeni formasyon şeklini alır. Buna bir örnek Şekil 5'te verilmiştir.

tir. Sürü kare formasyonu ile başlayıp sonrasında göreve devam ederken formasyon şeklini değiştirerek delta formasyonuna geçiş yapmıştır. Burada görüldüğü gibi sürü sistemi formasyon değişikliğini başarıyla gerçekleştirmiştir. Şekil 6'de sürü formasyonun genişleyerek devam etmesinin bir örneğinin simülasyonu aynı zamanda bir sinüoidal gezineyi izlerken verilmiştir. Bu da formasyonun şeklinin boyutunun üst seviye kontrolcümüz kullanılarak değiştirilmesine bir örnektir.



Şekil 5: Dron sürüsü üzerinde formasyon sürüsünün değişiminin a) 3B'de görünümü ve b) üstten görünümü.



Şekil 6: Dron sürüsü üzerinde formasyon şeklinin boyutunun değiştirilmesi a) 3B'de görünümü ve b) üstten görünümü.

3.2. Gazebo Simülasyonları

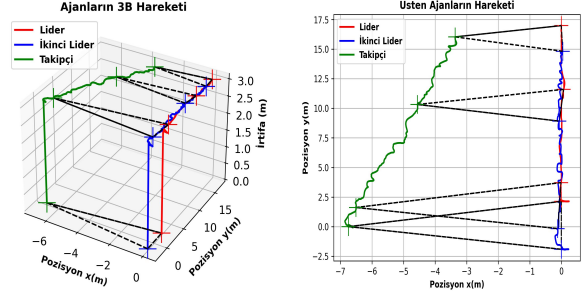
Burada Gazebo'da yapılan sürü formasyon şeklinin daralması simülasyon sonucu verilmiştir. Gazebo'dan aldığımız İHA'ların pozisyon verileri kullanılarak figürleştirilmiştir. Şekil 7'de sürü formasyonunun daralması görülmektedir.

4. Sonuçlar

Bu çalışmada bir İHA sürüsü için dağıtık bir kontrolcü ile sürü formasyon şeklinin değiştirilmesi, boyutlarının ayarlanması ve daha önceki çalışmamızda olduğu gibi sürünün formasyon şeklinin korunması yapılmıştır. Burada amaç gerçekte uygulanmasını test etmek için gerekliliklerin bir testinin yapılarak simülasyon ortamlarında sonuçlarının karşılaştırması ve incelemesi olmuştur. Görülmüşürki kısıtlı kamera görüş açısı formasyon görevlerinde etkilidir. Farklı görevlerde daha büyük görüş açılı kameraların kullanılması gerekmektedir. Kısıtlı kamera açısına (yatayda 60°) rağmen elde edilen sonuçlar doğrultusunda sürü sistemimiz oldukça tatmin edicidir.

5. Kaynakça

- [1] M. Dorigo, M. Birattari, S. Garnier, H. Hamann, M. M. de Oca, C. Solnon, and T. Stützle, "Swarm intelligence,"



Şekil 7: Gazebo'da gerçekleştirilen dron sürüsünün daralması hareket etmesinin a) 3B'de görünümü ve b) üstten görünümü.

Scholarpedia, vol. 2, no. 9, p. 1462, 2007.

- [2] S. Yuan, H. Wang, and L. Xie, "Survey on localization systems and algorithms for unmanned systems," *Unmanned Systems*, vol. 9, no. 02, pp. 129–163, 2021.
- [3] X. Cai and M. de Queiroz, "Dynamic formation control of multi-agent systems using rigid graphs," in *Dynamic Systems and Control Conference*, vol. 46186. American Society of Mechanical Engineers, 2014, p. V001T14A003.
- [4] T. H. Summers, C. Yu, S. Dasgupta, and B. D. Anderson, "Control of minimally persistent leader-remote-follower and coleader formations in the plane," *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2778–2792, 2011.
- [5] B. D. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Systems Magazine*, vol. 28, no. 6, pp. 48–63, 2008.
- [6] M. De Queiroz, X. Cai, and M. Feemster, *Formation control of multi-agent systems: a graph rigidity approach*. John Wiley & Sons, 2019.
- [7] M. A. Toksöz, S. Oğuz, and V. Gazi, "Decentralized formation control of a swarm of quadrotor helicopters," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*. IEEE, 2019, pp. 1006–1013.
- [8] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers," *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–26, 2021.
- [9] J. Ulrich, A. Alsayed, F. Arvin, and T. Krajinik, "Towards fast fiducial marker with full 6 dof pose estimation," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 2022, pp. 723–730.
- [10] B. Fidan, V. Gazi, S. Zhai, N. Cen, and E. Karataş, "Single-view distance-estimation-based formation control of robotic swarms," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5781–5791, 2012.
- [11] A. Tayebi and S. McGilvray, "Attitude stabilization of a vtol quadrotor aircraft," *IEEE Transactions on control systems technology*, vol. 14, no. 3, pp. 562–571, 2006.